Lab 6. Linear time series models

Let us first play around acf, pacf and the Ljung and Box test.

```
> x <- arima.sim(100,model=list(ar=c(.8,-.6), ma=c(0.7,0.4)))
             #simulate a TS with length 100
> par(mfrow=c(2,2))
> tsplot(x)  # plotting the series
> ?acf       # help on acf
> acf(x,40)  # acf plot
Autocorrelation matrix:
   lag       x
 1   0  1.0000
 2   1  0.1906
 3   2 -0.4748
> acf(x,40,"partial") #PACF plot
> xacf <- acf(x,40,plot=F)  # computing ACF without plot
> xacf <- xacf$acf[2:11]^2  #square autocorrelation 1:10
> Q <- 100*102*sum(xacf/(100-1:10))  #Box-Ljung test
> 1-pchisq(Q,10)    #computing P-value
[1] 0.00005613359
```

Let now play around the efficient market hypothesis. First, let us load the .xls file for SP500.

```
> SP500
           Date    Open    High     Low   Close     Volume Adj..Close.
   1 01/30/2004 1133.06 1133.20 1127.81 1131.13 1309560064     1131.13
   2 01/29/2004 1130.06 1134.19 1122.41 1134.11 1631110016     1134.11
   3 01/28/2004 1146.31 1149.08 1126.61 1128.48 1483849984     1128.48
   4 01/27/2004 1154.38 1155.14 1144.05 1144.05 1379830016     1144.05
> sp500 <- rev(SP500[,5])   #extracting closing prices
> sp500
   [1]  134.22  132.93  133.88  134.88  134.20  138.54  138.72  138.82
   [9]  138.69  141.77  142.72  141.81  140.00  139.57  139.95  137.39
>  sp500 <- log(sp500)         #log-prices
>  acf(sp500,400)                  # examining log-prices
>  n <- length(sp500)          # length of series
   ## calculating correlation without drift; here are two methods
>  x <- sp500[1:(n-1)]
>  y <- sp500[2:n]
>  sum(x*y)/sum(x^2)  # correlation without drift
```

```
[1] 1.00006
>  lsfit(x,y, intercept=F)$coef #here is a short cut.
       X
 1.00006
  #### caculating the drift with drifts ####
>  lsfit(x,y)$coef
 Intercept         X
 0.00218512 0.9997106
```

Here is an example for fitting an ARMA model to the daily log-returns of the SP500 stocks.

```
> return500 <- 100*diff(sp500)
> mean(return500)   #mean daily return
[1] 0.03986333
> return500 <- return500-mean(return500)  ### substract the mean out
> return500 <- as.ts(return500)    ###create a time series structure
> acf(return500)                    ## acf plot
> acf(return500, lag.max=100, type="partial")  ## pacf plot
> ?arima             #see what kind of helps are there
> return500.arima  <- arima.mle(return500, list(order=c(1,0,1)))
> return500.arima
Call: arima.mle(x = return500, model = list(order = c(1, 0, 1)))
Method:  Maximum Likelihood
Model :  1 0 1

Coefficients:
    AR : -0.33794
    MA : -0.35862

Variance-Covariance Matrix:
          ar(1)     ma(1)
ar(1) 0.2992435 0.2967183
ma(1) 0.2967183 0.2943774

Optimizer has  converged
Convergence Type: relative function convergence
AIC: 15962.01405
> arima.diag(return500.arima)   #arima diagnostics
> fore500 <- arima.forecast(return500, return500.arima$model, 10)
  ### 10 step forward prediction
>
```

```
> fore500
$mean:
5348: -1.203471e-002  4.067045e-003 -1.374429e-003  4.644786e-004
5352: -1.569673e-004  5.304598e-005 -1.792652e-005  6.058139e-006
5356: -2.047305e-006  6.918724e-007
 start deltat frequency
  5348      1         1

$std.err:
5348: 1.076341 1.076571 1.076598 1.076601 1.076601 1.076601 1.076601
5355: 1.076601 1.076601 1.076601
 start deltat frequency
  5348      1         1
### you may want to try the function ar.
### mean and arma coefficients can be simultaneously fitted as follows
>  return500 <- 100*diff(sp500)
>  armafit <- garch(return500~arma(1,1))
>  summary(armafit)
> summary(armafit)

Call: garch(formula.mean = return500 ~ arma(1, 1))

Mean Equation: return500 ~ arma(1, 1)

Conditional Variance Equation:  ~ garch(0, 0)

Conditional Distribution:  gaussian

------------------------------------------------------------

Estimated Coefficients:
------------------------------------------------------------
         Value Std.Error   t value Pr(>|t|)
    C  0.05649  0.025247    2.2374  0.01265
AR(1) -0.29831  0.466929   -0.6389  0.26146
MA(1)  0.30952  0.464647    0.6661  0.25267
    A  1.10191  0.005125  214.9860  0.00000

------------------------------------------------------------

AIC(4) = 15976.33
```

```
BIC(4) = 16002.66

Normality Test:
------------------------------------------------------------------
 Jarque-Bera P-value
      399108       0

Ljung-Box test for standardized residuals:
------------------------------------------------------------------
 Statistic  P-value Chi^2-d.f.
     29.55 0.003261         12

Ljung-Box test for squared standardized residuals:
------------------------------------------------------------------
 Statistic P-value Chi^2-d.f.
     374.8       0         12

Lagrange multiplier test:
------------------------------------------------------------------
 Lag 1 Lag 2  Lag 3 Lag 4   Lag 5  Lag 6 Lag 7 Lag 8 Lag 9 Lag 10 Lag 11
Lag 12     C
 9.464 2.395 -1.607 9.417 -0.1164 -2.152 2.459  1.98 -1.45 0.1588 0.8724
-0.008431 6.849

  TR^2 P-value F-stat    P-value
 283.9       0  27.26 4.415e-007
```

For the airline model, try the following example

```
> al.mod <- list(list(order=c(0,1,1)), list(order=c(0,1,1), period=12))
> fit <- arima.mle(ship, model=al.mod)
Call: arima.mle(x = ship, model = al.mod)
Method:  Maximum Likelihood
Model :

Coefficients:

Variance-Covariance Matrix:
               ma(1)          ma(2)         ar(12)         ma(12)
 ma(1)  1.405628e-002  5.203302e-003 -3.081360e-006 -3.160769e-006
 ma(2)  5.203302e-003  1.405628e-002 -8.454751e-007 -8.618194e-007
ar(12) -3.081360e-006 -8.454751e-007  7.283371e-001  7.473533e-001
```

```
ma(12) -3.160769e-006 -8.618194e-007  7.473533e-001  7.803968e-001

Optimizer has  converged
Convergence Type: relative function convergence
AIC: 1219.18261
```